הרצאה 25: אלגוריתמי קירוב

Textbook: Cormen, Leiserson, Rivest, Stein. Introduction to Algorithms.

מרצה: אורי שטמר

תזכורת:

בעיית קבוצה דוקרת (Hitting Set)

מופע: אוסף של n קבוצות $S_1, \dots, S_n \subseteq \{1, \dots, k\}$.

פתרון חוקי: קבוצת איברים $J \subseteq \{1, \dots, k\}$ כך שבכל קבוצה S_i בקלט יש לפחות איבר אחד $j \in S_i$ כך ש- $j \in J$. קבוצה כזו נקראת קבוצה דוקרת.

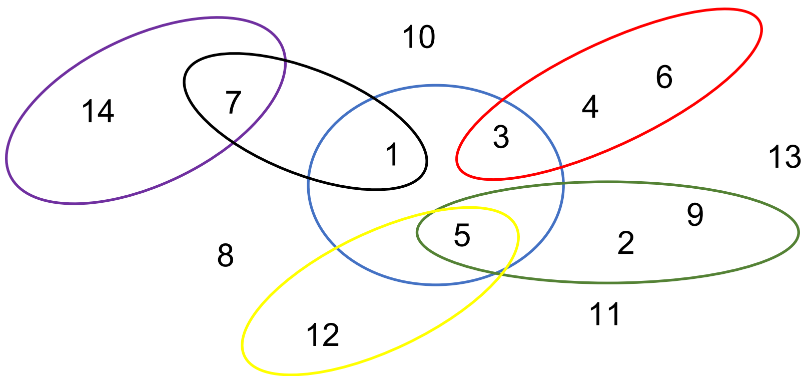
יש למצוא: פתרון חוקי J בגודל $|J|$ קטן ביותר.

בהרצאה הקודמת ראינו אלגוריתם קירוב לבעייה זו המבוסס על תכנון לינארי + אלגוריתם עיגול. היום נראה אלגוריתם קירוב חמדן לבעייה זו.

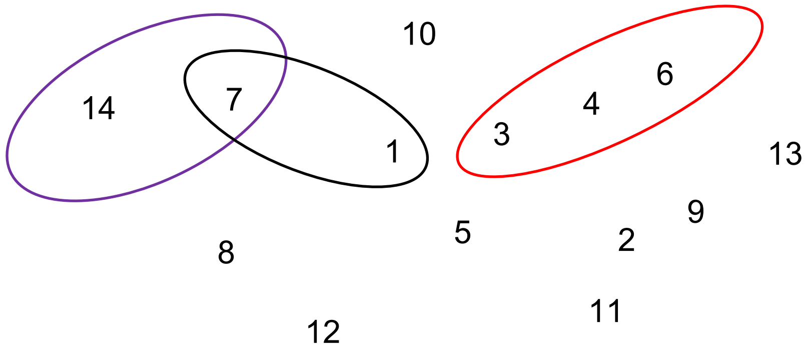
Algorithm Greedy-Hitting-Set

1. נאתחל $U = \{1,2, \dots, n\}$ ו- $J = \emptyset$
(כאן J מייצג את הפתרון שאנו בונים ו- U מייצג את הקבוצות S_u שעדיין לא דקרנו)
2. כל עוד $U \neq \emptyset$ נבצע:
   א. נבחר איבר $j \in \{1, \dots, k\}$ אשר דוקר מספר גדול ככל האפשר של קבוצות מבין הקבוצות שעדיין לא דקרנו. כלומר j הממקסם את $|\{u \in U : j \in S_u\}|$.
   ב. נמחק מ- U את האינדקסים של הקבוצות שנדקרות על ידי j.
   כלומר, $U \leftarrow U \setminus \{u : j \in S_u\}$
   ג. נוסיף את j לפתרון שאנו בונים. כלומר נבצע $J \leftarrow J \cup \{j\}$
3. נחזיר את J

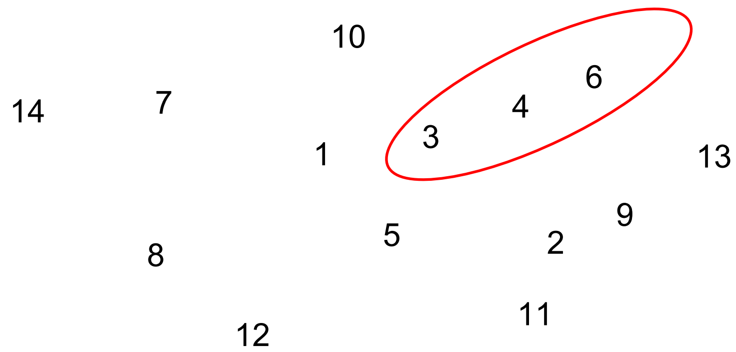
--- דוגמת ריצה ---



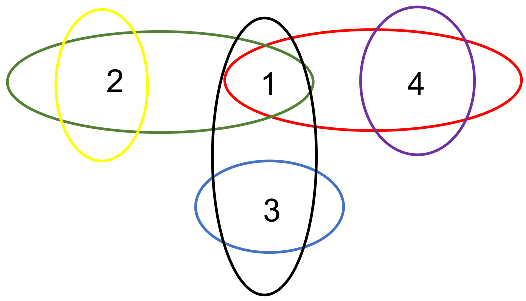
נבחר את 5 כי הוא דוקר מספר גדול ביותר של קבוצות (3 קבוצות). נמחק קבוצות שנדקרות ע"י 5 ונקבל



נבחר את 7 ונמחוק קבוצות שנדקרות ע"י 7. נקבל:



נבחר למשל את 4. סה"כ קיבלנו קבוצה דוקרת בגודל 3: הקבוצה *{5,7,4}*.



דוגמה עבורה האלגוריתם לא מוצא פתרון אופטימלי:
גודל פתרון אופטימלי הוא 3.
האלג' יחזיר פתרון בגודל 4.

<u>משפט:</u> האלגוריתם הנ״ל הוא אלגוריתם קירוב-$O(\log n)$ לבעיית קבוצה דוקרת.

<u>הוכחה:</u>
יהי $J_{OPT} \subseteq \{1, \dots, k\}$ פתרון אופטימלי עבור המופע שלנו. עלינו להראות שהאלגוריתם מחזיר פתרון J חוקי כך שמתקיים

$$|J| \leq O(\log n) \cdot |J_{OPT}|$$

<u>חוקיות:</u> כאשר האלגוריתם עוצר מקיים $U = \emptyset$, כאשר U מייצגת את הקבוצות S_u שעדיין לא דקרנו (אינדקס u נמחק מ- U רק כאשר אנו מוסיפים לפתרון איבר j כך ש- $j \in S_j$). לכן, בסיום הריצה, הפתרון המוחזר הוא חוקי.

שאלה: מדוע האלגוריתם בהכרח עוצר?
תשובה: נניח שאין קבוצות ריקות בקלט (אחרת אין פתרון למופע). הנחה זו היא בלי הגבלת הכלליות כי אנחנו יכולים לבדוק זאת בקלות לפני הרצת האלגוריתם ובמקרה שיש קבוצה ריקה בקלט נחזיר מיד "אין פתרון". כעת, בכל איטרציה אנו מכניסים איבר j לתוך J שדוקר לפחות קבוצה אחת שעדיין לא דקרנו. לכן אחרי לכל היותר n איטרציות נדקור את כל הקבוצות ואז יתקיים ש- $U=\emptyset$ והאלגוריתם יעצור.

<u>יחס קירוב:</u>
נשים לב שבכל איטרציה של האלגוריתם אנחנו מוסיפים בדיוק איבר אחד לפתרון שאנו בונים J. לכן, כדי לחסום את $|J|$ מספיק לחסום את מספר האיטרציות במהלך הריצה.

סימון: $U_\ell =$ הקבוצה U בתחילת האיטרציה ה- ℓ. בתחילת האיטרציה הראשונה מתקיים $U_1=\{1,\dots,n\}$.

נתבונן בתחילת האיטרציה ה ℓ (עבור ℓ כלשהו).

נזכור שהפתרון האופטימלי J_{OPT} דוקר כל קבוצה מתוך $S_1,\dots,S_n$, ובפרט דוקר כל קבוצה מבין הקבוצות שעדיין לא דקרנו עם הפתרון שאנו בונים, כלומר דוקר כל קבוצה שהאינדקס שלה ב- U_ℓ. לפי עיקרון שובך היונים, קיים איבר $i\in J_{OPT}$ אשר דוקר לפחות $\frac{|U_\ell|}{|J_{OPT}|}$ קבוצות כאלו.
מכיוון שהאלגוריתם שלנו בוחר איבר $j\in\{1,\dots,k\}$ אשר דוקר מספר גדול ככל האפשר של קבוצות שעדיין לא נדקרו, אנו יודעים שהאיבר הנבחר j דוקר לפחות $\frac{|U_\ell|}{|J_{OPT}|}$ קבוצות כאלו. לכן,

$$|U_{\ell+1}| \le |U_\ell| - \frac{|U_\ell|}{|J_{OPT}|} = |U_\ell|\cdot\left(1-\frac{1}{|J_{OPT}|}\right)$$

לכן, כל איטרציה מקטינה את $|U_\ell|$ (כלומר את מספר הקבוצות שעדיין לא דקרנו) בפקטור כפלי של $\left(1-\frac{1}{|J_{OPT}|}\right)$. לכן, לאחר ℓ איטרציות נקבל

$$|U_{\ell+1}| \le |U_1|\cdot\left(1-\frac{1}{|J_{OPT}|}\right)^{\ell} = n\cdot\left(1-\frac{1}{|J_{OPT}|}\right)^{\ell}$$

כמה איטרקציות נוכל לבצע לפני שהאלגוריתם יעצור?

נניח שהאלגוריתם מבצע t איטרציות. אזי בתחילת האיטרציה ה t מתקיים $|U_t|\ge 1$ ולכן

$$1 \le |U_t| \le n\cdot\left(1-\frac{1}{|J_{OPT}|}\right)^{t-1} \le n\cdot\left(e^{-1/|J_{OPT}|}\right)^{t-1} = n\cdot\exp\left(-\frac{t-1}{|J_{OPT}|}\right)$$

כאשר אי-השוויון האחרון נובע מהנוסחה $1+x\le e^x$ המתקיימת לכל $x\in\mathbb{R}$.

נוציא *ln* משני האגפים ונקבל

$$0 \le \log(n) - \frac{t-1}{|J_{OPT}|}$$

כלומר

$$t \le |J_{OPT}|\cdot\log(n)+1$$

זה מראה שמספר האיטרציות שהאלגוריתם מבצע, ולכן גם גודל הפתרון שהוא מחזיר, הוא לכל היותר $O(|J_{OPT}|\cdot\log(n))$.

מ.ש.ל.

בעיית הסוכן הנוסע עם אי-שוויון המשולש

מוטיבציה: סוכן מכירות רוצה לבקר ב- n ערים. הסוכן רוצה למזער את עלויות הנסיעה שלו.

אנחנו ממדלים את הבעייה הזאת בעזרת גרף מלא (לא מכוון) עם n קודקודים עם משקולות על הקשתות. הקודקודים מייצגים את הערים ומשקלי הקשתות מייצגים את עלות הנסיעה הישירה בין שני הערים המתאימים לקודקודי הקשת. פורמלית,

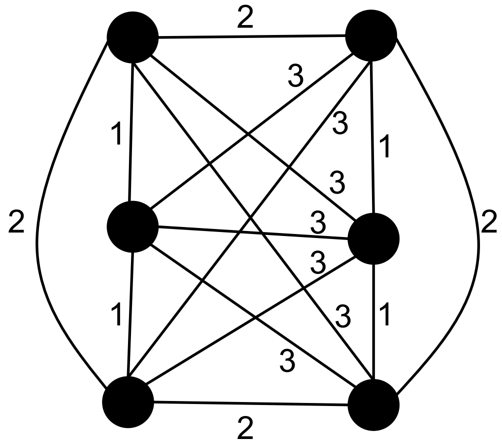
<u>מופע:</u> (G, c) כאשר
$G = (V, E)$ הוא גרף מלא ולא-מכוון,
$c: V \times V \to \mathbb{N}$ היא פונק' משקל אי-שלילית על הצלעות, כך שלכל $u, v, w \in V$ מתקיים
$$c(u, w) \leq c(u, v) + c(v, w)$$

<u>פתרון חוקי:</u> מעגל המילטוני בגרף. כלומר, מעגל פשוט שעובר דרך כל קודקוד בדיוק פעם אחת.

<u>יש למצוא:</u> מעגל המילטוני במשקל מינימום (כאשר משקל המעגל הוא סכום משקלי הקשתות במעגל).

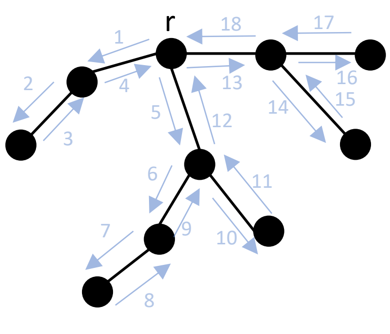
הערה: אפשר להצדיק את ההנחה ש- c מקיימת את אי-שוויון המשולש באופן הבא. ברוב מקרים, הדרך הזולה ביותר להגיע מעיר X לעיר Y היא על ידי נסיעה ישירה. במילים אחרות, אם במקום להגיע מ u ל w דרך v, ניסע ישירות מ u ל w אז זה רק יכול להקטין את עלות הנסיעה.

דוגמה:



<u>הגדרה:</u> יהי $T = (V, E)$ עץ. <u>מעגל-T</u> הוא מעגל (לא פשוט) ב- T שעובר דרך כל קשת בדיוק פעמיים.

<u>הערה:</u> בהינתן עץ T, ניתן למצוא מעגל-T בזמן ליניארי, למשל ע"י *DFS*.



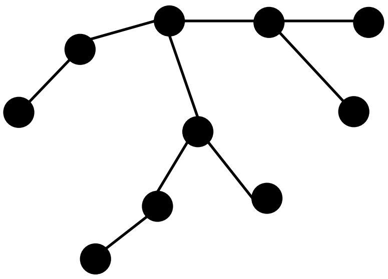
דוגמה: בהינתן עץ T, נקבע שרירותית שורש *r* ונתחיל ריצת *DFS* מ- *r*

Algorithm MTSP

1. נמצא *MST* של גרף הקלט G. נסמנו כ- T
2. נמצא מעגל-T של העץ T. נסמנו $P=\left(v_1,v_2,\dots,v_{2|V|-2},v_1\right)$
3. תהי $\left(v_{i_1},\dots,v_{i_{|V|}}\right)$ תת סדרה של P המכילה (בדיוק) את המופע הראשון של כל קודקוד ב- P.
כלומר, כדי לבנות את הסדרה הזאת הולכים לאורך המסלול P ומוסיפים לסדרה כל קוד' שנתקלים בו בפעם הראשונה.
4. נחזיר את המעגל ההמילטוני $P^*=\left(v_{i_1},\dots,v_{i_{|V|}},v_1\right)$.

דוגמת ריצה:

נניח גרף הקלט הוא הגרף המלא מעל הקודקודים הבאים, כאשר משקל כל קשת מסומנת היא 1 ומשקל כל הקשתות אשר לא מופיעות בציור הוא 2.



אזי בשלב 1, העפ"מ שנמצא יכיל בדיוק את הקשתות המסומנות.

נניח שבשלב 2 נמצא את המעגל-T שתואר בדוגמה האחרונה.
מה יהיה המעגל ההמילטוני שיוחזר בשלב 4?

<u>אבחנה 0:</u> האלגוריתם מחזיר מעגל המילטוני.

<u>הסבר:</u> מכיוון שהמסלול P (משלב 2) עובר דרך כל קשת בדיוק פעמיים, אז, בפרט, הוא עובר דרך כל קודקוד לפחות פעם אחת. לכן, לפי הגדרה, המעגל המוחזר בשלב 4 מכיל כל קודקוד בדיוק פעם אחת. מכיוון שגרף הקלט הוא גרף מלא, זהו מעגל המילטוני.

<u>משפט:</u> האלגוריתם הנ"ל הוא אלגוריתם קירוב-2 לבעיית הסוכן הנוסע עם אי-שוויון המשולש.

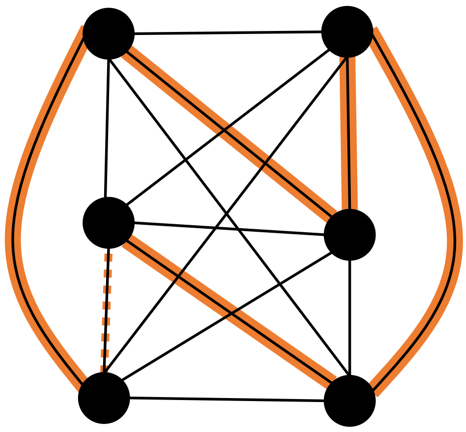
<u>סימונים:</u>

- נסמן ב *OPT* משקל פתרון אופטימלי עבור המופע שלנו
- נסמן ב w_{MST} את משקל ה *MST* משלב 1 באלגוריתם
- נסמן ב w_P את משקל המעגל P משלב 2
- נסמן ב w_{SOL} את משקל המעגל P^* המוחזר ע"י האלגוריתם בשלב 4

<u>טענה 1:</u> $w_{MST}\le OPT$

<u>הוכחה:</u> נניח בשלילה שמתקיים $w_{MST}>OPT$. יהי C מעגל המילטוני בגרף במשקל OPT. נסמן ב E_C את קבוצת הצלעות המופיעות במעגל C. נבחין כי לכל צלע e במעגל C מתקיים שהגרף $(V,E_C\setminus\{e\})$ הוא עץ פורש של גרף הקלט G, ומשקלו הוא לכל היותר OPT. לכן, לפי ההנחה בשלילה, קיבלנו עץ פורש במשקל קטן ממש מזה של העפ"מ שלנו T. סתירה.

בציור: כל מעגל המילטוני, אם נוריד ממנו צלע כלשהי נקבל עץ פורש (במשקל קטן שווה ממנו כי המשקלים אי-שליליים). לכן משקל עפ"מ הוא קטן שווה ממשקל מעגל המילטוני קטן ביותר.



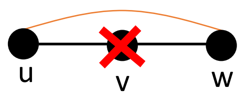
__אבחנה 2:__ $w_P = 2 \cdot w_{MST}$

__הסבר:__ המסלול P מכיל כל צלע של העפ"מ T בדיוק פעמיים.

__טענה 3:__ $w_{SOL} \leq w_P$

__הוכחה:__
נובע מכך שפונק' המשקל c מקיימת את אי-שוויון המשולש. ספציפית, המעגל המוחזר P^* מתקבל מ- P על ידי השמטת צמתים. מאי-שיוויון המשולש, כל השמטה של צומת v יכולה רק או לא לשנות או להקטין את המחיר.

בציור:



W(u,w)<=w(u,v)+w(v,w)

__הוכחת המשפט:__
לפי אבחנה 0, הפתרון המוחזר חוקי. לפי אבחנה וטענות *1,2,3* מתקיים

$$\begin{bmatrix}\text{משקל}\\ \text{הפתרון}\\ \text{המוחזר}\end{bmatrix} = w_{SOL} \leq w_P = 2 \cdot w_{MST} \leq 2 \cdot OPT$$

מ.ש.ל.